



UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office

Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

MF

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.
-----------------	-------------	----------------------	---------------------

08/950,760 10/15/97 WOLLRATH

A 06502.0063-0

EXAMINER

TM02/0313

JEFFREY A BERKOWITZ
FINNEGAN HENDERSON FARABOW GARRETT &
DUNNER
1300 I ST. NW
WASHINGTON DC 20005-3315

LAW,
ART UNIT PAPER NUMBER

2151

DATE MAILED:

03/13/01

16

Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner of Patents and Trademarks

Office Action Summary

Application No. 08/950,760	Applicant(s) Wollrath, et al
Examiner S. La	Group Art Unit 2151



Responsive to communication(s) filed on Jul 13, 2000

This action is FINAL.

Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under Ex parte Quayle, 1035 C.D. 11; 453 O.G. 213.

A shortened statutory period for response to this action is set to expire 3 month(s), or thirty days, whichever is longer, from the mailing date of this communication. Failure to respond within the period for response will cause the application to become abandoned. (35 U.S.C. § 133). Extensions of time may be obtained under the provisions of 37 CFR 1.136(a).

Disposition of Claim

Claim(s) 54-69 is/are pending in the application.

Of the above, claim(s) _____ is/are withdrawn from consideration.

Claim(s) _____ is/are allowed.

Claim(s) 54-69 is/are rejected.

Claim(s) _____ is/are objected to.

Claims _____ are subject to restriction or election requirement.

Application Papers

See the attached Notice of Draftsperson's Patent Drawing Review, PTO-948.

The drawing(s) filed on _____ is/are objected to by the Examiner.

The proposed drawing correction, filed on _____ is approved disapproved.

The specification is objected to by the Examiner.

The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. § 119

Acknowledgement is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d).

All Some* None of the CERTIFIED copies of the priority documents have been received.

received in Application No. (Series Code/Serial Number) _____.

received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

*Certified copies not received: _____

Acknowledgement is made of a claim for domestic priority under 35 U.S.C. § 119(e).

Attachment(s)

Notice of References Cited, PTO-892

Information Disclosure Statement(s), PTO-1449, Paper No(s). _____

Interview Summary, PTO-413

Notice of Draftsperson's Patent Drawing Review, PTO-948

Notice of Informal Patent Application, PTO-152

-- SEE OFFICE ACTION ON THE FOLLOWING PAGES --

DETAILED ACTION

1. Claims 54-69 are pending. This action is in response to the amendment filed 12/27/2000. Applicant has amended claim 67.
2. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.
3. Claims 54-69 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bezviner et al (US #5,613,148) in view of Edwards et al (US #5,901,315).

As to claim 58, Bezviner teaches a method in a data processing system (DSOM runtime) having associated objects (server instances), the method comprising:

- receiving from a remote source a request to access one of the objects (col 5, lines 7 - 11);
- determining whether the requested object is active (col 7, lines 38 - 64);
- activating the requested object when it is inactive (col 7, lines 38 - 64).

Bezviner does not teach (1) the requested object is in a group of objects, (2) activation of the requested object is in a virtual machine associated with the group of objects.

As to (1)-(2), Edwards teaches data processing method, including a requested object (Java code portion) in a group of objects (target application comprising both portions of Java code and portions of native C/C++ code), activating a virtual machine (launch a Java virtual machine) associated with the group of objects before activating the requested object in the virtual machine (then run the target application under Java VM). In the case where there is a virtual machine associated with (single-machine solution, host machine), VM activation is not performed before activating requested object (Java debugger API 20). See col. 2, lines 5-7, 21-24; col. 4, lines 34-34, 41-57, 65-67; col. 5, lines 1-8, 36-55; col. 9, lines 13-39.

As to determining whether there is a virtual machine associated with the group of objects when it has been determined that the requested object is inactive, this is inherent to Edwards because (a) the group of objects include active objects (portions of code being debugged which requires execution) and inactive objects (portions of code not being debugged), and (b) the system of Edwards needs to identify the Java code portions before invoking Java debugger class (which is provided by Java VM). In other words, the system needs to identify the Java VM associated with Java code portions.

Since Bezviner identifies operating in different environments (col. 2, lines 1-37) and Edwards provides a mechanism to do so, it would have been obvious to combine the teachings, which meets the claimed limitations.

As to claim 59, Bezviner as modified by Edwards teaches accessing the activated object (Bezviner: service provided by a server object instance) (Edwards: debugging a Java code portion).

As to claim 60, Bezviner as modified by Edwards teaches Java virtual machine (Java virtual machine) (discussion of claim 58).

As to claim 54, note the rejection of claim 58, in particular discussion of Edwards with respect to activating/launching a VM before activating a requested object under the activated VM. As to at least one other object in the group remains inactive, it is met by parts of the Java code of the target application not being currently debugged since debugging is typically performed line by line or block by block.

As to claims 55 and 57, note discussion of claims 59 and 60, respectively.

As to claim 56, it is covered by claim 58.

As to claims 61-64, these are the program product claims of claims 54-57, respectively. Note corresponding claims for rejections.

As to claims 65-66, these are the program product claims of claims 58-59, respectively. Note corresponding claims for rejections.

As to claim 67, it is covered by claim 54 except for and also met by Bezviner as modified by Edwards: first computer (Bezviner: client machine; Edwards: debugger client), second computer (Bezviner: server machine; Edwards: target machine), process on first machine (client), object activator on second machine (Bezviner: DSOM modified by Edwards: javaprobe) (discussion of claim 58).

As to claim 68, it is covered by claim 58 and further note discussion of claim 67 for first computer, second computer and object activator. Locating the VM is inherent to activating it or determining it is already associated with the objects (ie, active).

As to claim 69, it is an apparatus claim of claim 54. Note rejection of claim 54.

4. Applicant's arguments filed 12/27/2000 have been fully considered but they are not persuasive.

4.1. Regarding claims 54, 58-59, applicant argued neither Bezviner nor Edwards teaches determining whether there is a virtual machine associated with the group of objects when it has been determined that the requested object is inactive, and cited reason as that in conventional systems, there is no virtual machine activated associated with the group when there is an inactive object in the group, thus no reason to determine whether. (Page 3, 1st-2nd paragraphs).

The examiner's position is as follows. As to determining whether there is a virtual machine associated with the group of objects when it has been determined that the requested object is inactive, this is an inherent step to Edwards because (a) the group

(target application) of objects include active objects (portions of code being debugged, which receives debugging instructions) and inactive objects (portions of code not being debugged) at any time, and (b) the system of Edwards needs to identify the Java parts of the code portions before invoking Java debugger class provided by Java VM. In other words, the system needs to identify the Java programming paradigm associated with Java code portions, ie., to determine whether there is Java VM associated with an inactive object (portions of code to be debugged).

The system of Edwards inter-operates between two programming environments, represented by debugging a target application which comprises portions of Java code (which requires Java VM to run and debug) and portions of C/C++ code (which does not require Java VM to run and debug). When the Java code portions are debugged, the Java VM is first activated/launched before activating the object (Java code portion to be run and debugged). In the case where the Java VM is already associated with the object (ie, already activated) in the host machine which runs Java VM (single machine solution), VM activation is not performed before activating the object.

As to applicant's repeatedly cited argument that in conventional systems, there is no virtual machine activated associated with the group when there is an inactive object in the group, thus no reason to determine whether, it is not clear what the conventional systems are, how the conventional systems relate to the present invention and the claim language, how the conventional systems relate to the prior art relied on, and how this argument is supported by the application as filed or by the prior art in the field in general. If the present invention as claimed does not apply to conventional systems, this aspect has not brought out in the claims. If the argued conventional systems refer to systems before Java was introduced, this argument does not disqualify the prior art relied on. Further, in conventional systems such as Windows which operates DOS VMs and system VM, there are DOS VMs associated with DOS objects/applications, and system VM associated with a Windows object/application, regardless of whether there is inactive objects in the group (DOS or Windows). The association is represented by the type of environment an application was written for (DOS or Windows). A typical definition of inactive is a state

which is not affected by the current command, which is met by a portion of code in Edwards not currently being debugged. The teaching of conventional systems such as Windows with DOS VMs and system VM may be found in standard books on MS Windows operating system. The typical definition of an active/inactive state may be found in Computer Dictionary by Microsoft Press. The argument is thus not persuasive.

4.2. Regarding claims 58-60, 65-66 and 68, applicant argued that the prior art does not teach activating the requested object in the virtual machine associated with the group of object when it is determined that there is a virtual machine associated with the group of objects because conventional systems do not activate objects of a group individually and as a result do not have a virtual machine for the group and an inactive object in the group. (Page 4, 1st-2nd paragraphs).

The examiner disagrees. As to activating the requested object, this is met by the combination of Bezviner and Edwards, as discussed in the rejection of claim 58 and further explanation was also provided in section 6 of the previous office action. As to the argument that conventional systems do not activate objects of a group individually and as a result do not have a virtual machine for the group and an inactive object in the group, refer to section 4.1. for the discussions of conventional systems and inactive object. Further, to one of ordinary skill in the art, to active is to bring to an active state which is a state being able to receive current command. Bring an object / portion of code into debugging state enables object / portion of code to receive debugging command, therefore, activating the object. If applicant's activating differs from the typical understanding, this aspect has not been brought out by the claim language. In addition, in conventional systems such as Windows which operates DOS VMs and system VM, objects in a group are activated individually as in the cases where a Windows spreadsheet program is active (taking input) while a word processor is inactive (not taking input), or one cell in the spreadsheet program is active and another cell remains inactive. In all such cases, the system VM is associated with the group of objects (Window applications) to provide underlying Window services. The argument is thus not persuasive.

4.3. Regarding claim 54, applicant argued the prior art does not teach at least one other object of the group remains inactive because "The debugging of code in Edwards et al does not activate an object of the code". (page 5, 1st paragraph).

Refer to section 4.2. for discussion of at least one other object of the group remains inactive. As to the argument of "does not activate an object of the code", it is not clear what was the intended meaning.

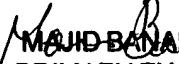
5. THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for response to this final action is set to expire THREE MONTHS from the date of this action. In the event a first response is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event will the statutory period for response expire later than SIX MONTHS from the date of this final action.

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sue Lao whose telephone number is (703) 305-9657. A voice mail service is also available at this number. The fax phone number for the organization where this application or proceeding is assigned is (703) 308-9051 for regular communications and After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the Group receptionist whose telephone number is (703) 305-9600.

Sue Lao
February 27, 2001


MAUD BANANKHAH
PRIMARY EXAMINER